COMPOSER 2.0

Nils Adermann
@naderman

Jordi Boggiano
@seldaek

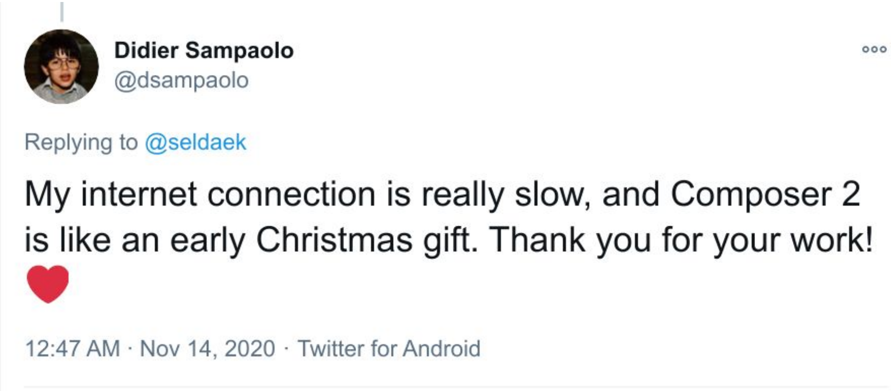
Private Packagist
https://packagist.com

SymfonyWorld
ONLINE 2020
DECEMBER 3-4

Composer 2.0 was released October 24, 2020

- Development began summer 2018
- 28 code contributors
- Many more helpers reporting issues, commenting, and testing pre-releases
- Made possible by Private Packagist and its customers

Thank you!



Didier Sampaolo
@dsampaolo

Replying to @seldaek

My internet connection is really slow, and Composer 2 is like an early Christmas gift. Thank you for your work! ❤️

12:47 AM · Nov 14, 2020 · Twitter for Android

Goals for 2.0

- Performance Improvements
- Better reproducibility
    - Most serious 1.x bugs are edge cases which are difficult to debug and hard to reproduce
- Better error reporting
- New features which become easier to add by BC breaks/refactoring

- Keep upgrading as painless as possible

Why 2.0 at all and not 1.x?

Improving Performance

- What makes Composer slow?
  - I/O
    - Network
      - Metadata JSON downloads
      - Package file downloads
    - Memory access
      - Writing, accessing and modifying GBs of memory
  - CPU
    - Sequential unpacking of code archives

Improving Performance

What does Composer use memory for?

- JSON representation of every version of every package that may fit your requirements
- representation of dependencies/conflicts for SAT solver between all of these packages

Solutions

- Reduce number of package versions which “may fit my requirements”
- Represent dependencies/conflicts more efficiently

Reduce number of package versions which “may fit my requirements”

- **Composer 1** lazy loads packages while creating memory representation of dependencies
  - Idea: Solver only loads what it needs when it gets to that point
  - Problems
    - Solver just waits for same info at a later point
    - Impossible to reduce set of packages before generating dependencies
    - Parallelized network access becomes hard to manage

Composer 2.0 refactors process into multiple clearly separated steps:

- Recursively download metadata only for package versions which may really get installed
- Reduce number of package versions in memory as far as possible
- Generate solver memory representation of dependencies

=> BC Break (for plugins) => 2.0

Represent dependencies/conflicts more efficiently

SAT Solver takes boolean expressions, e.g.

```
foo/bar 1.0 requires baz/qux ^2.0             (- foo/bar 1.0 | baz/qux 2.0.0 | baz/qux 2.0.1 | baz/qux 2.1.0)
foo/bar 1.0 conflicts with baz/qux ^2.0       (- foo/bar 1.0 |- baz/qux 2.0.0) & (- foo/bar 1.0 |- baz/qux 2.0.1) &
                                              (- foo/bar 1.0 |- baz/qux 2.1.0)
```

You can only install one version of a package
=> automatically generate a conflict for each pair of versions

```
foo/bar 1.0, 1.1, 1.2                         (- foo/bar 1.0 |- foo/bar 1.1) & (- foo/bar 1.0 |- foo/bar 1.2) &
                                              (- foo/bar 1.1 |- foo/bar 1.2)
```

Extreme Growth $\binom{n}{2} = \frac{n!}{2(n-2)!}$

				Symfony	
	3 versions	6 versions	100 versions	500 versions	1000 versions
Composer 1	3 rules	15 rules	4,950 rules	**124,750** rules	**499,500** rules
Composer 2	1 rule	1 rule	1 rule	1 rule	1 rule

Composer 2.0 uses a special single multi conflict rule representation for all of these rules

```
foo/bar 1.0, 1.1, 1.2                         oneof(foo/bar 1.0, foo/bar 1.1,foo/bar 1.2)
```

Improving Performance: Network

- JSON Metadata & Package archive downloads
  - **Parallelization** of HTTP requests with curl multi
  - Use of **HTTP/2** features to reduce server round-trips
  - More reliable and feature complete than Composer 1 plugin implementations (hirak/prestissimo, symfony/flex) which were limited by plugin interface
- Packagist.org protocol improvements
  - Reduced amount of data transferred
  - Improved cacheability by splitting dev versions from releases
  - Stability improvements to packagist.org infrastructure

Note: Improvements require ext-curl (usage stats: installed on 97% of composer user systems)

Improving Performance: Archive Extraction

- Composer 2.0 unzips all downloaded archives in parallel
  - Requires Linux/OS X/WSL
  - Requires “unzip” CLI command in $PATH

Improving Performance

- What makes Composer slow?
  - I/O ✓
    - Network ✓
      - Metadata JSON downloads ✓
      - Package file downloads ✓
    - Memory access ✓
      - Writing, accessing and modifying GBs of memory ✓
        - Reduce number of package versions which “may fit my requirements” ✓
        - Represent dependencies/conflicts more efficiently ✓
  - CPU ✓
    - Sequential unpacking of code archives ✓

Benchmarks

- https://blog.packagist.com/composer-2-0-is-now-available/ by Jordi - @seldaek

  Updates: **60% less time**

- https://susi.dev/composer2-perf by Susey - @sasunegomo

  Installs: 44% less time

  Updates: 80% less time 81% less memory (1.15GB to 244MB)

- https://developers.ibexa.co/blog/benchmarks-of-composer-2.0-vs-1.10

  by Jani Tarvainen - @velmu

  Installs: 27% less time

  Updates: 50% less time 72% less memory (2.5GB to 711MB)

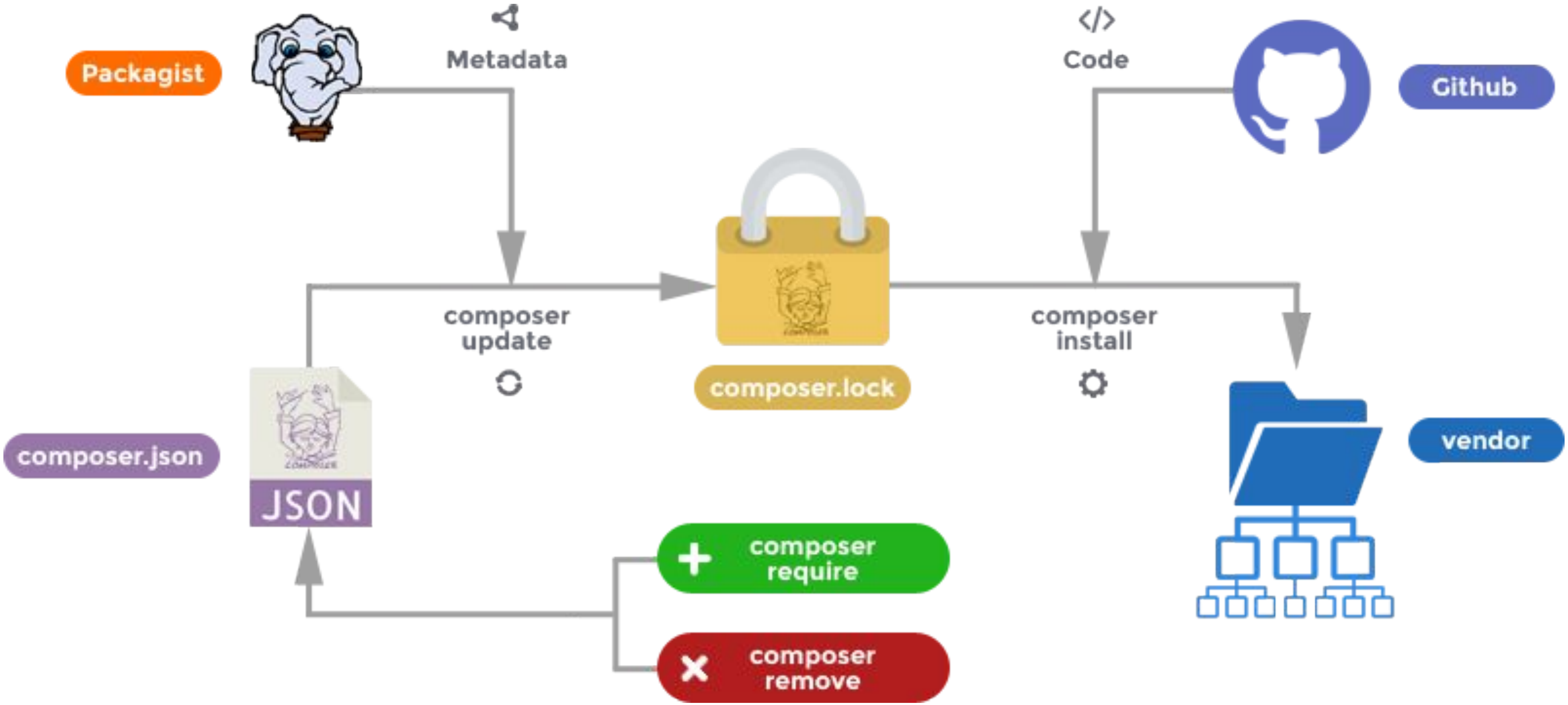
  Partial Update (composer require):

  23% less time **88% less memory (648MB to 81MB)**

Better Reproducibility:

composer update
vs
composer install

Separating update & install



Separating update & install

vendor			
	symfony/http-foundation:	5.1.8	previous local upgrade attempt
composer.lock			
	symfony/http-foundation:	4.4.16	old production state
composer.json			
	symfony/http-foundation:	5.0.*	limited upgrade for now, because of 5.1 issues

```
naderman@saumur:~/projects/composer/test/symfony-http-foundation$ composer update
Loading composer repositories with package information
Updating dependencies
Lock file operations: 0 installs, 1 update, 0 removals
 - Upgrading symfony/http-foundation (v4.4.16 => v5.0.11)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 3 installs, 1 update, 1 removal
 - Removing symfony/deprecation-contracts (v2.1.3)
 - Installing symfony/polyfill-php72 (v1.17.0): Extracting archive
 - Installing symfony/polyfill-intl-idn (v1.17.1): Extracting archive
 - Installing symfony/mime (v5.1.8): Extracting archive
 - Downgrading symfony/http-foundation (v5.1.8 => v5.0.11): Extracting archive
Generating autoload files
6 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
```

New Features

Ignoring specific platform requirements

Trying to test your project on PHP8?

```
composer update --ignore-platform-reqs
```

Installs on PHP8

May install packages requiring PHP extensions you do not have

```
composer update --ignore-platform-req=php
```

Installs on PHP8

Checks all extension requirements as usual

On deploy always run:

```
composer check-platform-reqs
```

Partial Updates to specific versions

```
// composer.json
    "require": {
        "symfony/http-foundation": “^4.0 || ^5.0",
    }

// composer.lock
      "packages": [{
            "name": "symfony/http-foundation",
            "version": "v4.4.16",

$ composer update symfony/http-foundation:5.0.*
Loading composer repositories with package information
Updating dependencies
Lock file operations: 0 installs, 1 update, 0 removals
 - Upgrading symfony/http-foundation (v4.4.16 => v5.0.11)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 0 installs, 1 update, 0 removals
 - Downloading symfony/http-foundation (v5.0.11)
 - Upgrading symfony/http-foundation (v4.4.16 => v5.0.11): Extracting archive
```

Repository Priorities

- Repositories are canonical by default:
  - First repository which has a package for a given name wins
  - to restore old behavior of merging package versions use `“canonical”: false`

- Limit packages a repository can provide

```
{
    “type”: “composer”,
    “url”: “https://some-third-party.com/composer-repo/”,
    “only”: [“foo/*”, “bar/baz”],
    “exclude”: [“foo/qux”]
}
```

Upgrading your projects

Best Case

```
composer self-update --2
composer update / composer install
```

No errors, everything works as before.

Upgrading issues with plugins

```
foo/bar requires composer-plugin-api ^1.0.0 -> no matching package found.
```

- Update foo/bar if new version with Composer 2.0 support available
- Contact author of foo/bar plugin
- Temporarily remove the requirement for the plugin to test

symfony/flex is compatible as of 1.9.8!

https://github.com/symfony/flex/pull/617

ocramius/package-versions

- Compatible with Composer 2.0 as of 1.8.0
  - requires PHP ^7.4
- `composer require` **`composer/package-versions-deprecated`**
  - We forked the package, now compatible with PHP ^7.0
  - replaces ocramius/package-versions
    => satisfies all requirements of ocramius/package-versions

- Building new code requiring runtime access to package info?

  Runtime Composer Utilities https://github.com/composer/composer/blob/master/doc/07-runtime.md
  - Automatically autoloaded in every Composer project

```
\Composer\InstalledVersions::isInstalled('vendor/package'); // returns bool

use Composer\Semver\VersionParser;
\Composer\InstalledVersions::satisfies(new VersionParser, 'vendor/package', '2.0.*');
```

Autoloading Issues

Check deprecation warnings in Composer 1

```
Class Foo\Bar located in ./src/SomeName/Bar.php does not comply with psr-4 autoloading
standard.
It will not autoload anymore in Composer v2.0. in
phar:///usr/local/bin/composer/src/Composer/Autoload/ClassMapGenerator.php:18
```

Make sure directories match class names as defined in PSR-0/4.

How far along is Composer 2.0?

Fast Adoption

- Composer 2.0 on packagist.org
  - **35%** of installs in November 2020
  - **20%** of updates in November 2020

Stable

- Current release 2.0.8
- Open issues are mostly uncommon edge cases
- Works well in production for large numbers of developers

Upgrade to Composer 2.0 now!

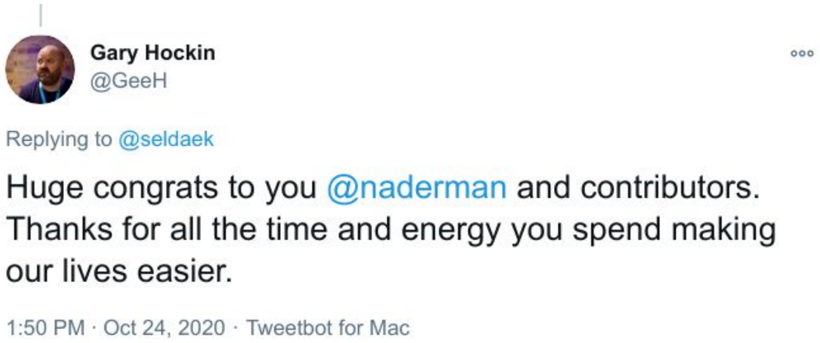
```
composer self-update --2
```

```
Updating to version 2.0.x (2.x channel).
  Downloading (100%)
Use composer self-update --rollback to return to version 1.10.17
```

Your lock file is commited, no risk involved in trying it locally!

Back to v1? `composer self-update --1`

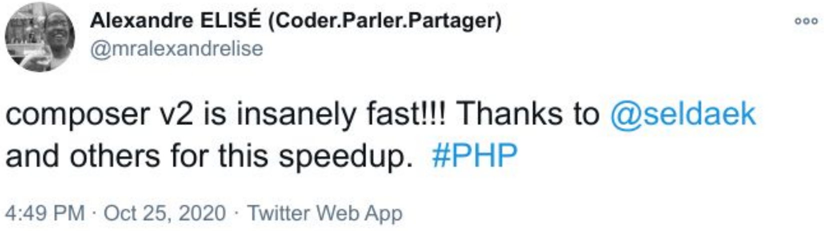
Feedback


Gary Hockin
@GeeH

Replying to @seldaek

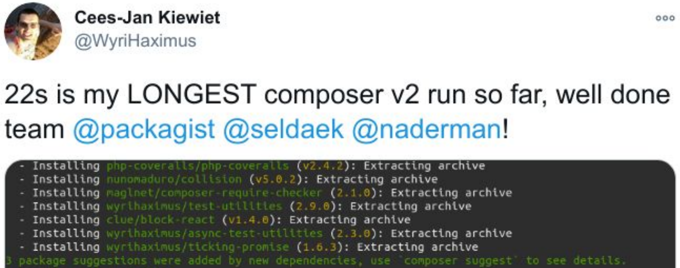
Huge congrats to you @naderman and contributors. Thanks for all the time and energy you spend making our lives easier.

1:50 PM · Oct 24, 2020 · Tweetbot for Mac


Alexandre ELISÉ (Coder.Parler.Partager)
@mralexandrelise

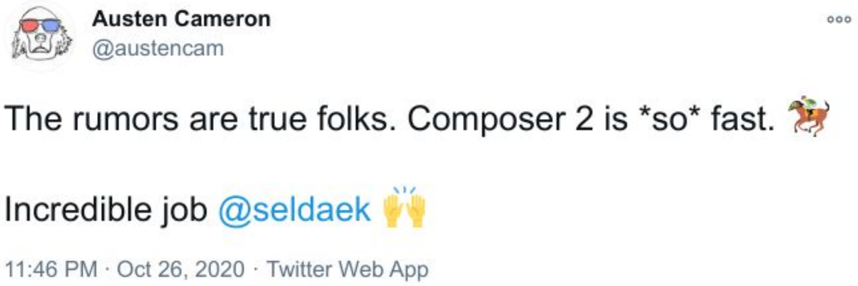
composer v2 is insanely fast!!! Thanks to @seldaek and others for this speedup. #PHP

4:49 PM · Oct 25, 2020 · Twitter Web App


Cees-Jan Kiewiet
@WyriHaximus

22s is my LONGEST composer v2 run so far, well done team @packagist @seldaek @naderman!

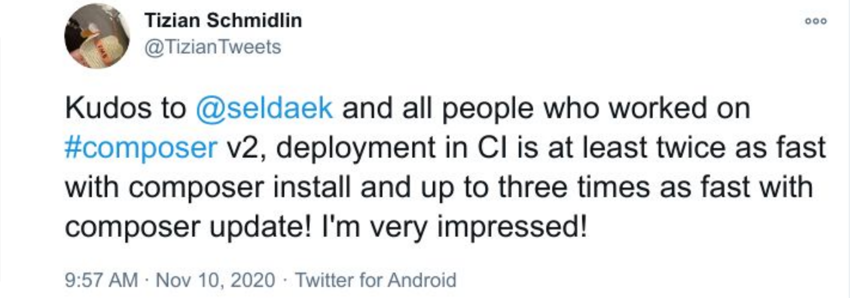
```
  - Installing php-coveralls/php-coveralls (v2.4.2): Extracting archive
  - Installing nunomaduro/collision (v5.0.2): Extracting archive
  - Installing maglnet/composer-require-checker (2.1.0): Extracting archive
  - Installing wyrihaximus/test-utilities (2.9.0): Extracting archive
  - Installing clue/block-react (v1.4.0): Extracting archive
  - Installing wyrihaximus/async-test-utilities (2.3.0): Extracting archive
  - Installing wyrihaximus/ticking-promise (1.6.3): Extracting archive
3 package suggestions were added by new dependencies, use `composer suggest` to see details.
```


Austen Cameron
@austencam

The rumors are true folks. Composer 2 is *so* fast. 🏇

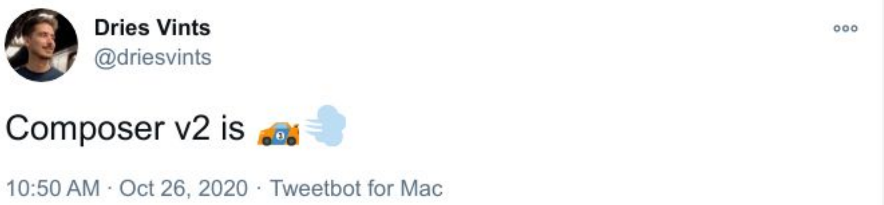
Incredible job @seldaek 🙌

11:46 PM · Oct 26, 2020 · Twitter Web App


Tizian Schmidlin
@TizianTweets

Kudos to @seldaek and all people who worked on #composer v2, deployment in CI is at least twice as fast with composer install and up to three times as fast with composer update! I'm very impressed!

9:57 AM · Nov 10, 2020 · Twitter for Android

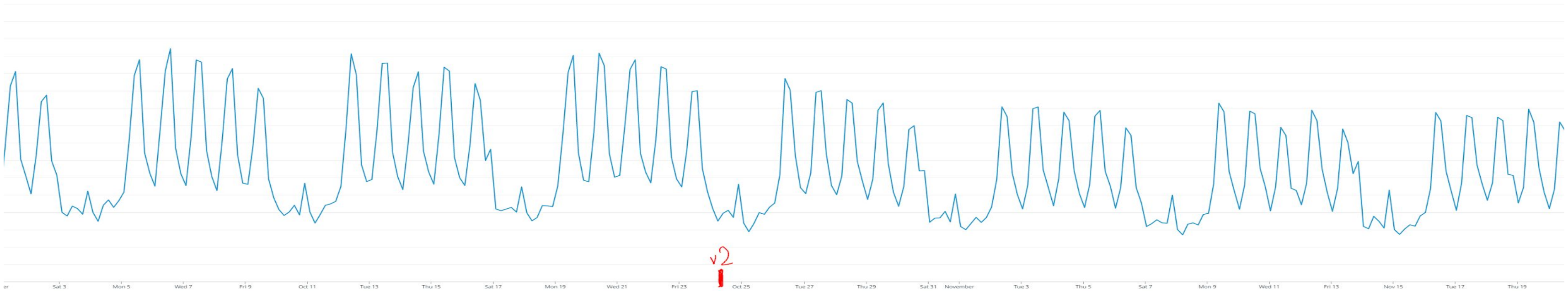

Dries Vints
@driesvints

Composer v2 is 🏎️💨

10:50 AM · Oct 26, 2020 · Tweetbot for Mac

packagist.org Bandwidth

- Composer 2 reduced bandwidth from 74 TB/week to 58 TB/week so far
  - We want everyone to upgrade Composer right now
    So Composer 2 supports PHP down to 5.3



What’s next?

- Requiring PHP 7.1 in Composer 2.2
  - Drop for external PHP5 support, internal refactorings only
  - Composer 2.1 will receive long term bug/security fixes

- Disabling Composer 1 support on packagist.org
  - Date depends on usage, please upgrade!

Resources

- Packagist Blog: Composer 2.0 is now available!
  https://getcomposer.org/2
- Changelog
  https://github.com/composer/composer/releases/tag/2.0.0
- Upgrade Guide
  https://github.com/composer/composer/blob/master/UPGRADE-2.0.md
- Composer Plugin Readiness for 2.0
  https://github.com/composer/composer/issues/8726

Questions / Feedback?

Private Packagist
https://packagist.com

E-Mail: contact@packagist.com
Twitter: @seldaek & @naderman